Abnormality Detection in Musculoskeletal Radiographs

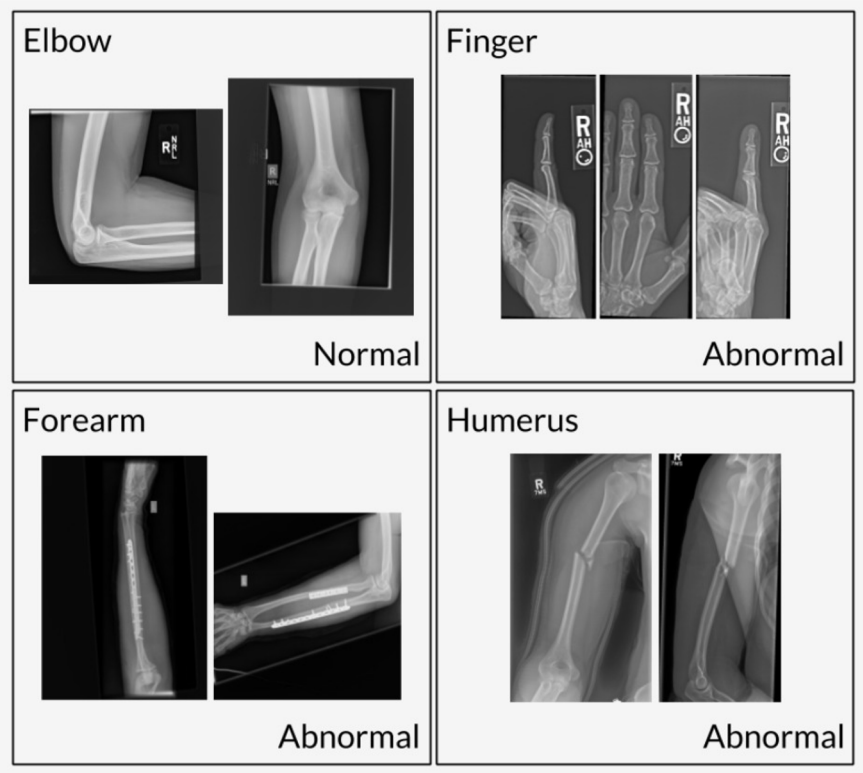
Divya Shankar | Sabrina Sok
Snigdha Jain

Introduction to MURA

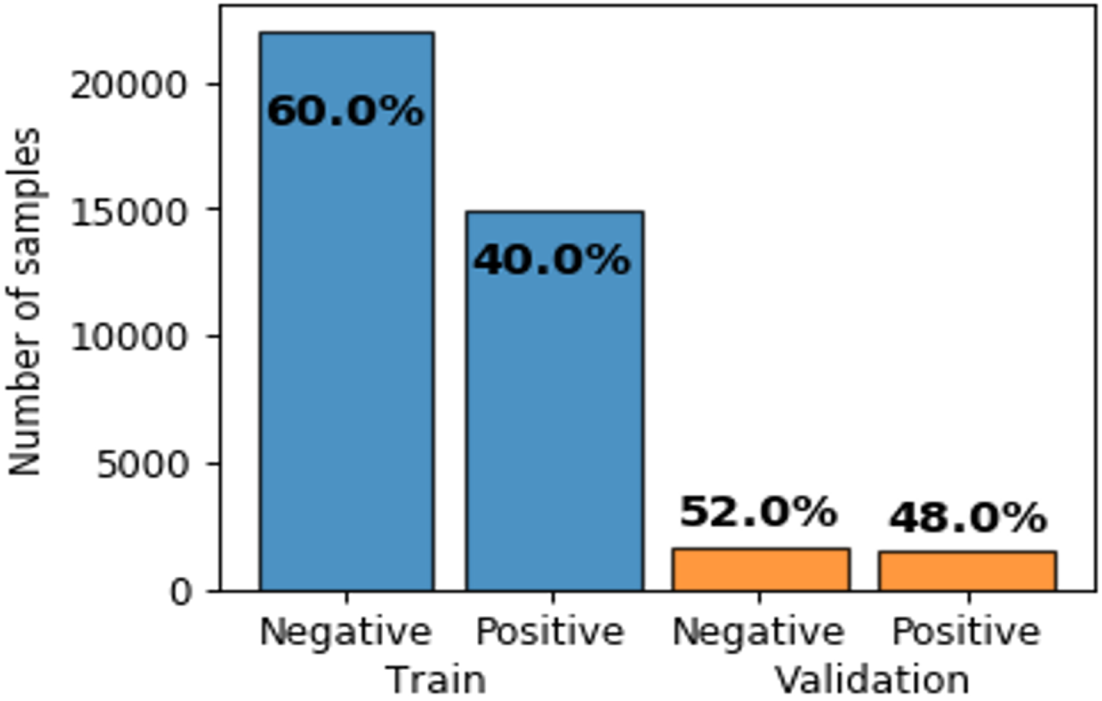
- Musculoskeletal Radiographs (MURA)
  - Musculoskeletal conditions affect more than 1.7 billion people worldwide
  - Most common cause of severe, long-term pain and disability, with 30 million emergency department visits annually and increasing.
  - Bone X-Rays are used to detect these musculoskeletal conditions.
  - Large dataset of Bone X-rays
- Problem Statement
  - Classify X-Rays from MURA dataset into:
    - Normal
    - Abnormal

Data Description

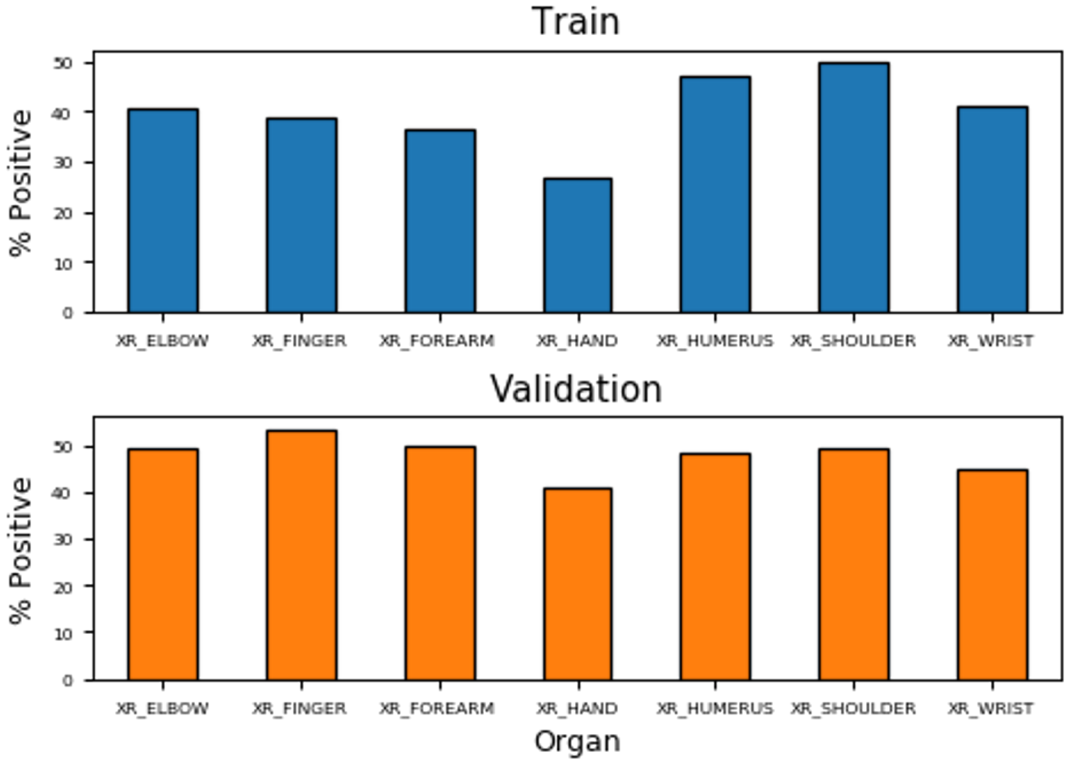
1. Dataset is created by Stanford.
2. Consists of 14,863 studies from 12,173 patients, with a total of **40,561 multi-view radiographic images**.
3. **Body parts covered:** elbow, finger, forearm, hand, humerus, shoulder, and wrist.
4. Study manually **labeled as normal or abnormal**
5. Image dimensions ~ **512*512**



Distribution of Data



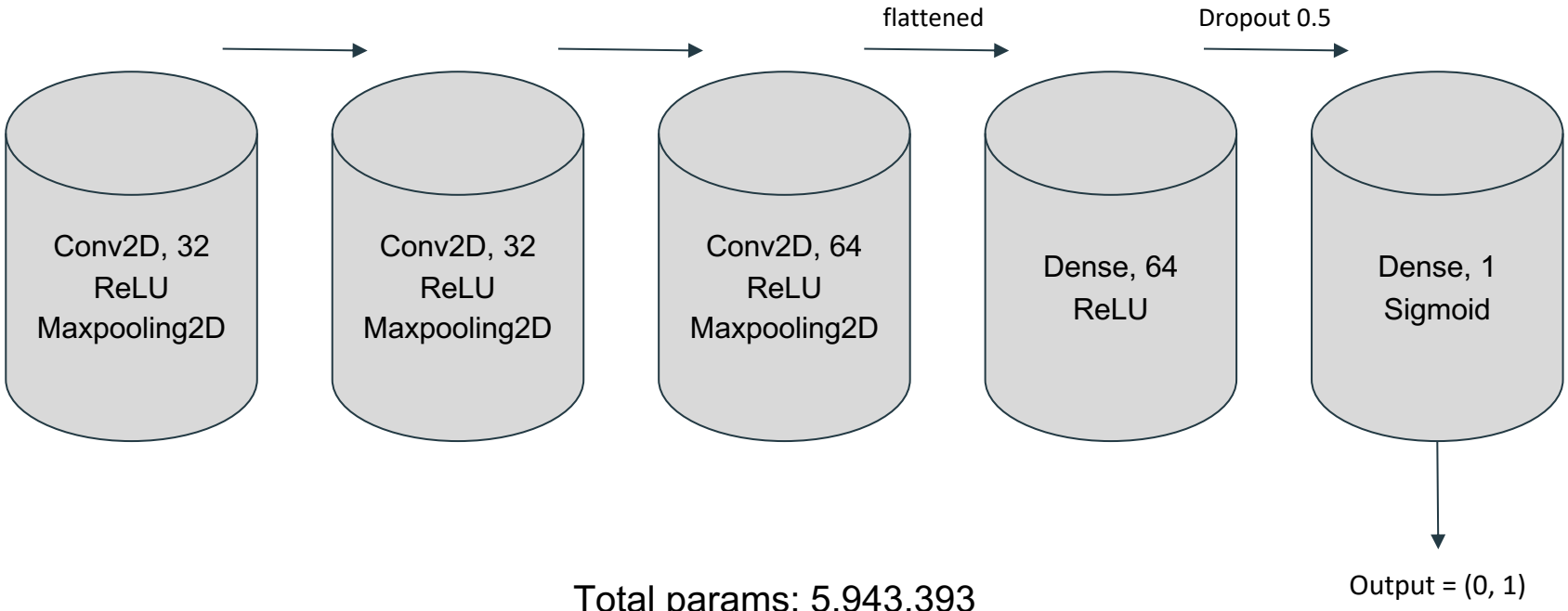
Distribution of Data



Project Approach

1. **Image Preprocessing:**
    - Images resized to 224*224 OR 300*300
    - ImageDataGenerator & flow_from_directory used to generate a batch of augmented/normalized data
2. **Models Trained:**
    - Simple CNN
    - Complex CNN
    - Transfer Learning: MobileNet
3. **Comparison of Models based on Accuracy**

Simple Model Architecture

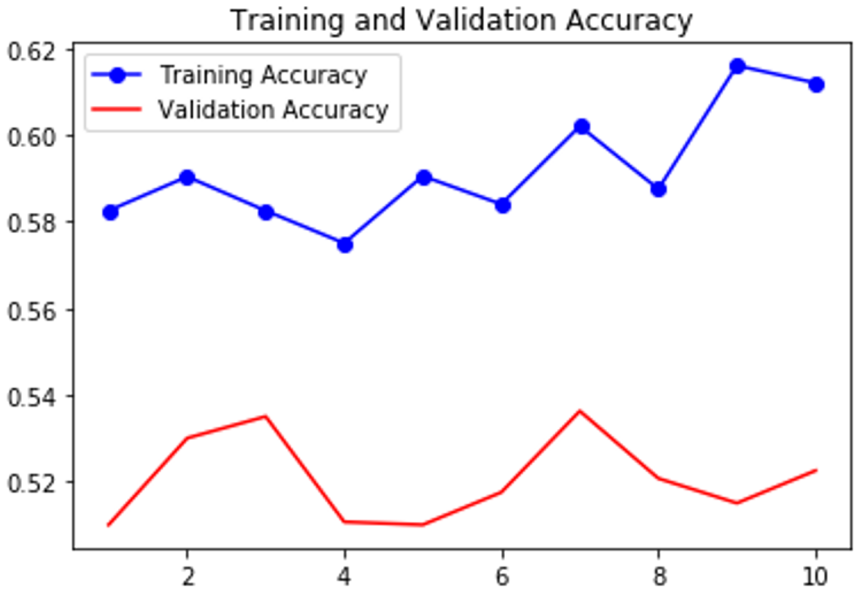


Total params: 5,943,393

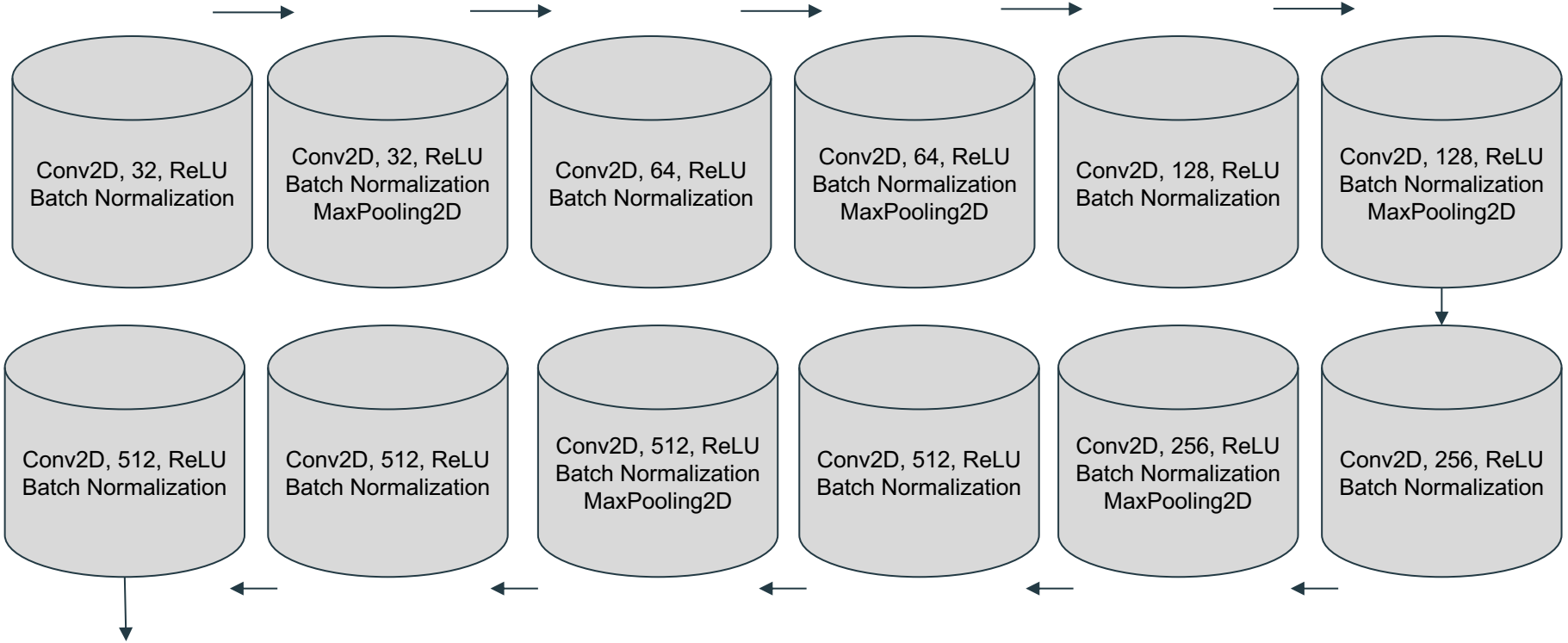
Simple Model Metrics

epoch	accuracy	loss	lr	val_accuracy	val_loss
0	0.5825	0.702807135	0.001	0.50999999	0.674242139
1	0.5903614	0.687252626	0.001	0.529999971	0.660216331
2	0.5825	0.683307526	0.001	0.535000026	0.763573766
3	0.575	0.677880373	0.001	0.510664999	0.681453049
4	0.5905	0.683911421	0.001	0.50999999	0.672540843
5	0.584	0.672008354	0.001	0.517499983	0.733212948
6	0.602	0.674156964	0.001	0.536249995	0.606750607
7	0.5875	0.673356977	0.001	0.52070266	0.698915899
8	0.616	0.667771111	0.001	0.514999986	0.729526162
9	0.612	0.66865963	0.001	0.522499979	0.691279471

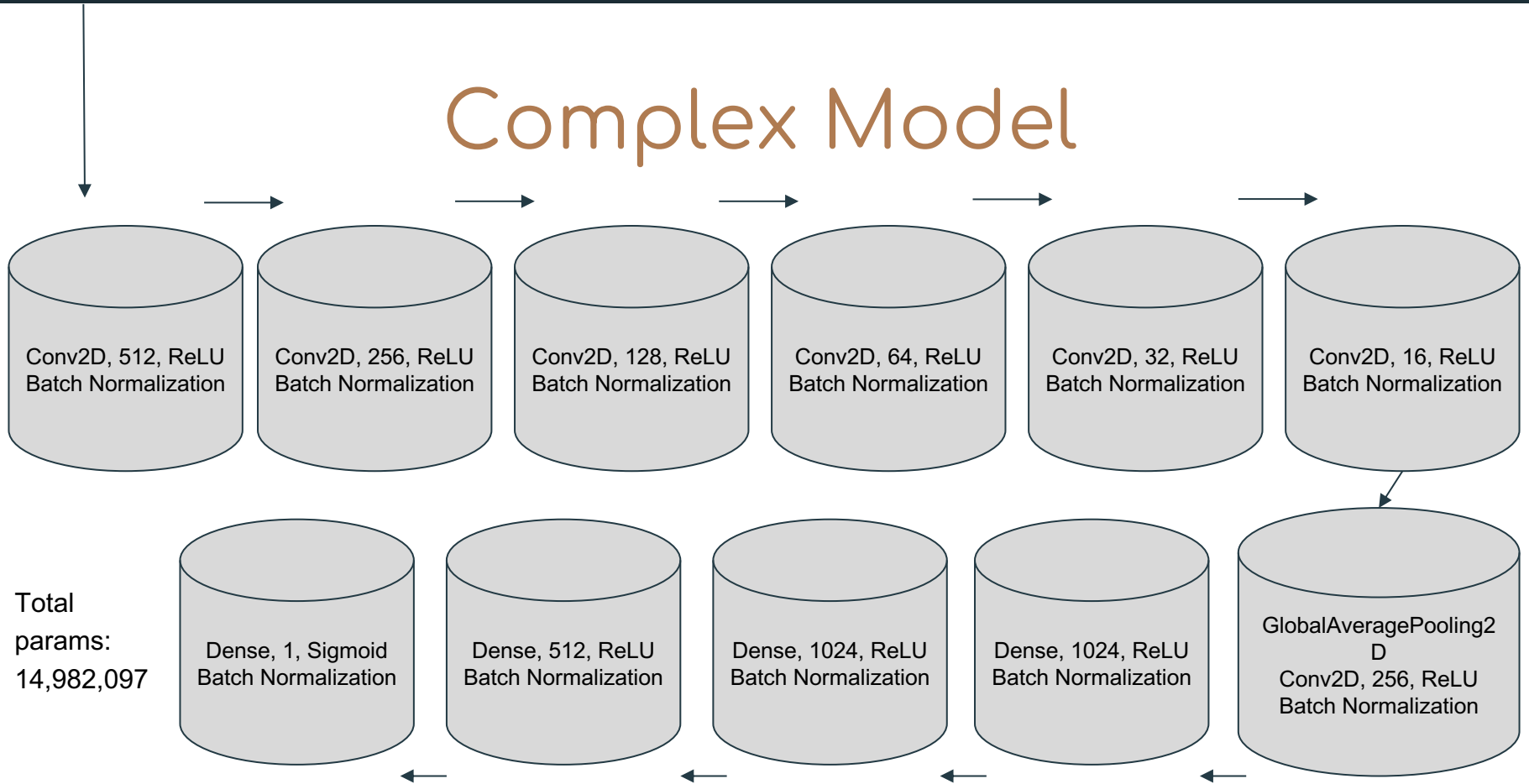
Simple Model Accuracy & Loss



Complex Model



Complex Model

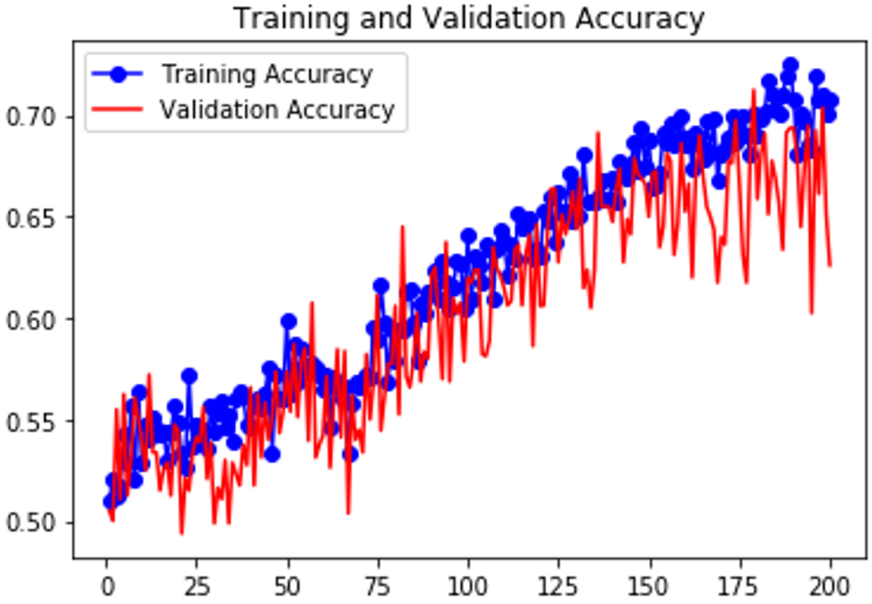


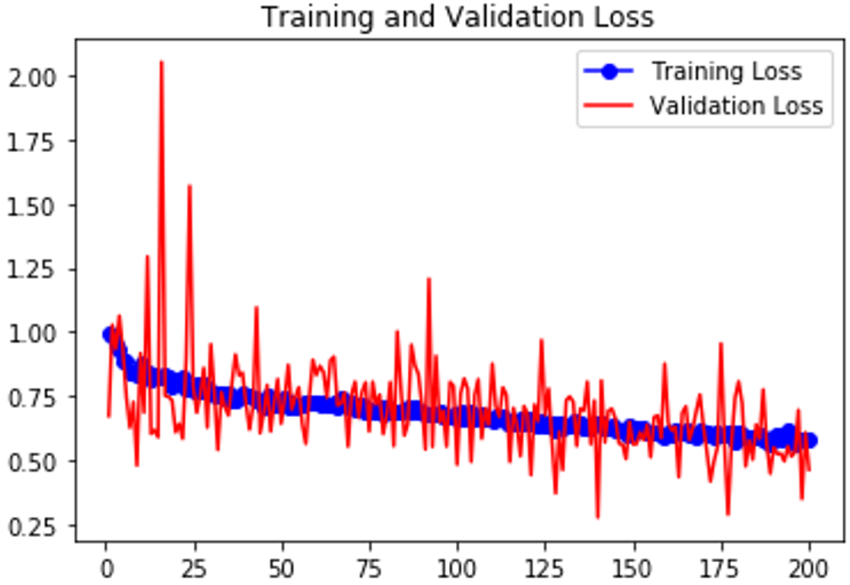
Total params: 14,982,097

Complex Model

epoch	accuracy	loss	lr	val_accuracy	val_loss
180	0.6985	0.596471215	2.50E-05	0.681249976	0.721990943
181	0.7015	0.588501526	2.50E-05	0.691250026	0.476119369
182	0.717	0.578701547	2.50E-05	0.651250005	0.577794135
183	0.7095	0.588868066	2.50E-05	0.677540779	0.502207994
184	0.7025	0.591975325	2.50E-05	0.668749988	0.639457047
185	0.7005	0.598578427	2.50E-05	0.65625	0.583139777
186	0.7105	0.581348874	2.50E-05	0.633750021	0.775833488
187	0.7195	0.5762399	2.50E-05	0.691342533	0.559177339
188	0.725	0.565876691	2.50E-05	0.693750024	0.447849989
189	0.708	0.589882666	2.50E-05	0.693750024	0.550647497
190	0.681	0.602205128	2.50E-05	0.678749979	0.523856401
191	0.701	0.584711273	2.50E-05	0.644918442	0.526481688
192	0.69779116	0.5956034	2.50E-05	0.675000012	0.495926708
193	0.686	0.6154132	2.50E-05	0.694999993	0.554816306
194	0.6835	0.594823656	2.50E-05	0.602500021	0.514467001
195	0.7195	0.567557128	2.50E-05	0.69259721	0.530557752
196	0.7075	0.579087101	2.50E-05	0.661249995	0.695942104
197	0.7095	0.582646748	2.50E-05	0.703750014	0.348975807
198	0.701	0.584441682	2.50E-05	0.651250005	0.608325422
199	0.708	0.57984041	2.50E-05	0.626097858	0.463423938

Complex Model





MobileNet

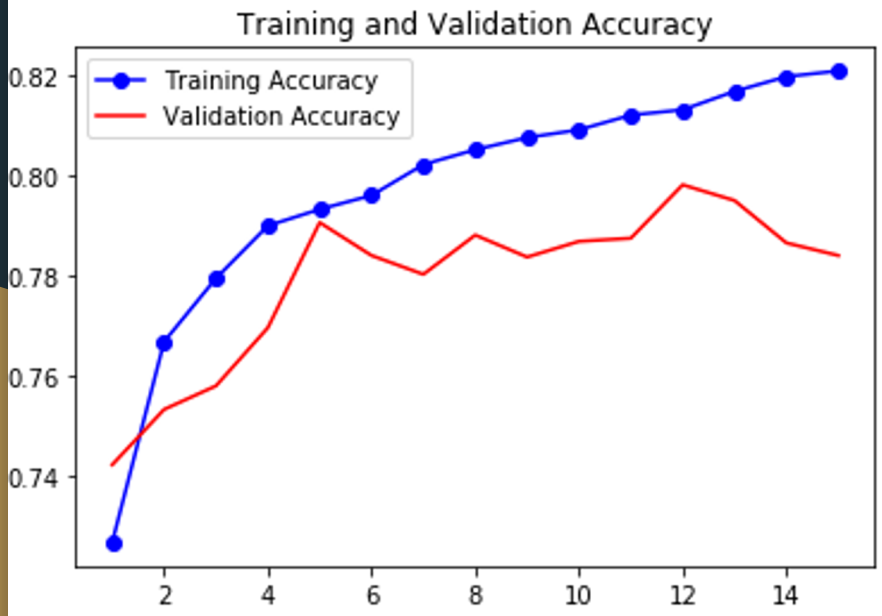
```
def mobileNet():
    model = keras.applications.mobilenet_v2.MobileNetV2(weights = 'imagenet',
                    include_top = False,input_shape = (224, 224, 3))
    x = model.output
    x = GlobalAveragePooling2D()(x)
    preds = Dense(1, activation='sigmoid')(x)
    model = Model(inputs=model.input, outputs=preds)
    model.compile(loss='binary_crossentropy', optimizer='Adam', metrics=['accuracy'])
    return model
```

Total params: 2,259,265

MobileNet Metrics

epoch	accuracy	loss	lr	val_accuracy	val_loss
0	0.7266899	0.552387251	1.00E-04	0.74216789	0.175077423
1	0.7667355	0.500884401	1.00E-04	0.75321418	0.320380628
2	0.77939576	0.482617973	1.00E-04	0.757917821	1.157320261
3	0.78988266	0.47077612	1.00E-04	0.769520223	0.282702774
4	0.79317	0.461015076	1.00E-04	0.790529966	0.716248274
5	0.7959411	0.456180468	1.00E-04	0.783944786	0.228243828
6	0.8021082	0.45095887	1.00E-04	0.780181885	0.29776147
7	0.80506957	0.444122456	1.00E-04	0.788021326	0.259390146
8	0.8074875	0.440826911	1.00E-04	0.783631206	0.221271217
9	0.8090361	0.436155514	1.00E-04	0.786767006	0.528053761
10	0.81197023	0.430409047	1.00E-04	0.787394166	0.221214056
11	0.81302977	0.428730231	1.00E-04	0.798055828	0.723202229
12	0.8167246	0.421415293	1.00E-04	0.794920027	0.673280895
13	0.81965876	0.420471713	1.00E-04	0.786453426	0.419653594
14	0.8207998	0.416761306	1.00E-04	0.783944786	0.756575346

MobileNet Accuracy & Loss

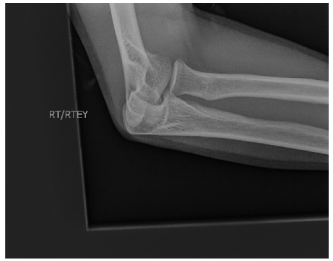


Comparison of Models

Model/Metric	Accuracy	Loss
Simple CNN	~61%	~67%
Complex CNN	~71%	~58%
MobileNet	~82%	~41%

Prediction

Wrong Prediction: Predicted as Abnormal



Right Prediction: Predicted as Abnormal

Conclusion

Abnormal

Normal

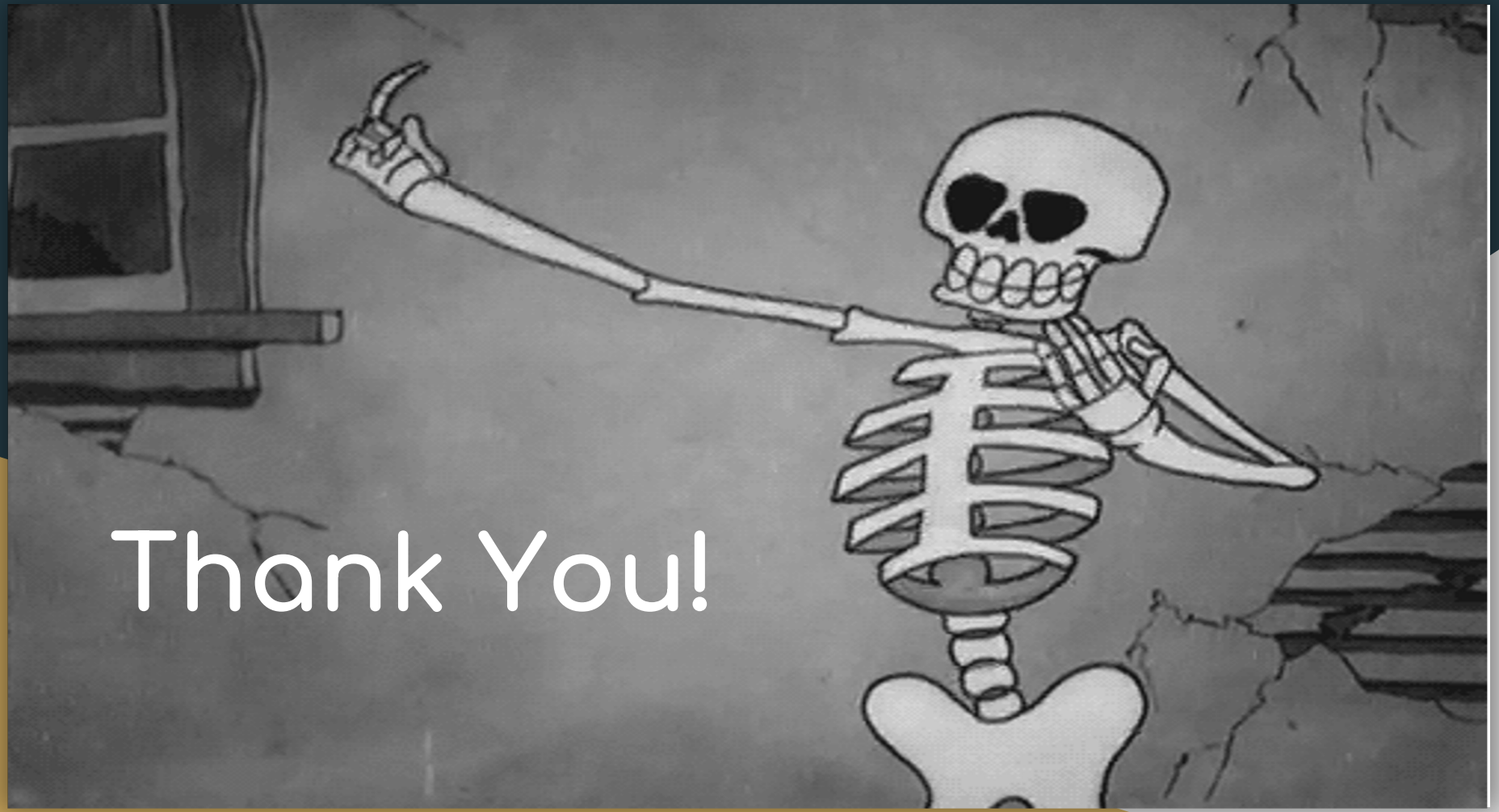
Conclusion

Normal

Abnormal

Future Work

- Improve model accuracy
- Extensive Hyperparameter tuning
- Implement regularization to avoid overfitting
- Deployment of solution on cloud platforms
- Development of a front-end application


Thank You!

References

1. MURA Dataset: https://stanfordmlgroup.github.io/competitions/mura/
2. Keras Documentation: https://keras.io